

## Web Server monitoring using AimBetter

For companies that work with Web-based applications, it is often difficult to monitor and troubleshoot errors, as they can't be tracked from the machine used by the user. More often than not, they are derived by web server and website settings at the host.

Web server monitoring tools designated for specific web servers and designed by the software companies that made the web servers only monitor web server activities and don't display application specific or web site specific parameters – that makes them short of being a comprehensive solution, not to mention they are not able to monitor hybrid applications, which are only partly web-based.

AimBetter, as a comprehensive tool that monitors operating system servers, database servers and web servers, polls data from all three and therefore can monitor applications whether they are locally installed, web-based or hybrid, and whether they use a database or not.

With AimBetter, you can monitor any application used by your organization, manage your app pools, and access all the monitoring tools provided from any machine connected to the internet, subject to permissions (at all levels, e.g., network level, machine level, server level, application level, user level, etc., as determined by the client).

AimBetter does not require local installation on the servers it monitors, but only a service to remotely communicate with. All you need to monitor your servers with AimBetter is to have it installed on one of your machines and it can be set to seamlessly communicate and monitor all your servers.

AimBetter's comprehensive approach provides IT personnel all the tools required to enhance their viewpoint, the information available to them, and their troubleshooting capabilities to those of expert DBAs.

AimBetter monitors all layers and aspects of a server, database, IIS, in one place, an all-in-one approach, enabling IT personnel monitor, maintain, and troubleshoot their servers from one place and by that reducing overhead, saving time on coordination, reducing TTR and reducing costs of operation.

AimBetter helps you manage your day more efficiently and view your OS server, DB server and Web server on one display, thanks to its comprehensive all-in-one approach.

The screenshot shows the AimBetter web monitoring interface. The top navigation bar includes 'Home', 'Performance', 'Queries', 'Observer', 'Web', and 'Connections'. The 'Web' tab is active. Below the navigation bar, there are tabs for 'Live' and 'History', and a date range selector set to 'Last 1 day ending 23-05-2022 14:53'. A filter is applied: 'Status = 404 Not Found, 500 Intern...'. The main table displays a list of requests with columns for Client IP, Url, Method, Start Time, Duration, Status, Host, Web Site, Server IP, and Port. One request is highlighted in yellow:

Client IP	Url	Method	Start Time	Duration	Status	Host	Web Site	Server IP	Port
172.17.0.108	/binGoApi/api/main/Del...	POST	23-05-2022 14:49	1 min	204 No Content	VM-TD-WEB	web.trellidor.co.il	185.127.9.22	443

Below the table, the 'Body' section shows the request details:

```
filename: "mrgPfdFiles_23549274.pdf"
```

On the right side of the body view, a summary table shows key details:

Url	/binGoApi/api/main/Deletefile	Client IP	172.17.0.108
Method	POST	Server IP	185.127.9.22
Status	204 No Content	Port	443
Host	VM-TD-WEB	Duration	1 min
Web Site Name	web.comp.co.il		

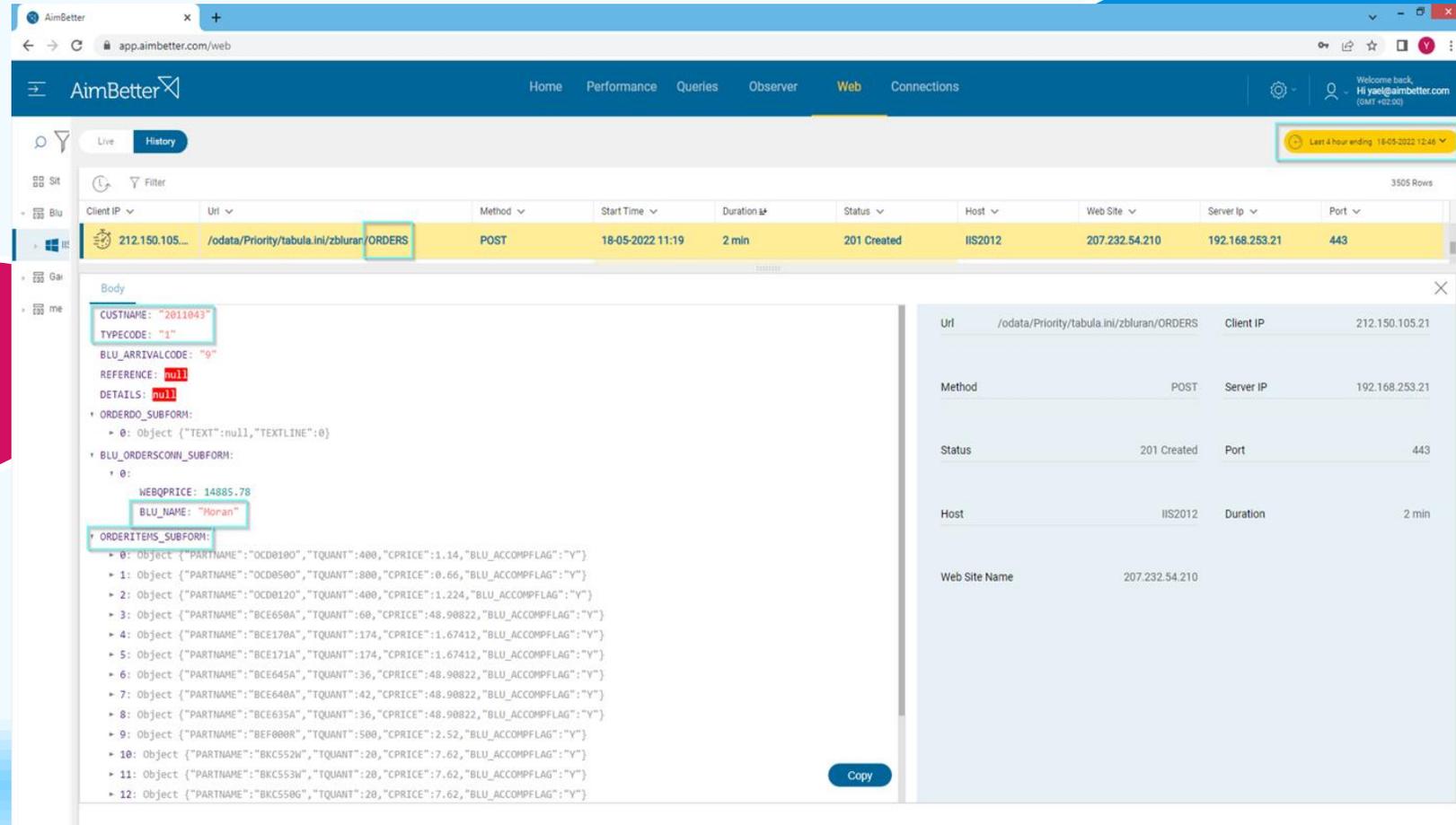
A 'Copy' button is located at the bottom right of the body view.

AimBetter monitors requests by time, method, and status (the last showing error messages if there are).

In the following screenshot, requests are filtered by status to show error statuses:

As can easily be seen, AimBetter logs all the request details and displays them in an orderly way:

Client IP, URL, method, start time, duration, status, host, web site, server IP, and port.



The screenshot shows the AimBetter web interface. The main content area displays a table of request records. One record is highlighted in yellow, and its details are shown in a modal window. The request body is displayed as JSON, and a summary table of request parameters is shown on the right side of the modal.

Client IP	Uri	Method	Start Time	Duration	Status	Host	Web Site	Server Ip	Port
212.150.105...	/odata/Priority/tabula.ini/zbluran/ORDERS	POST	18-05-2022 11:19	2 min	201 Created	IIS2012	207.232.54.210	192.168.253.21	443

```
Body
{
  "CUSTNAME": "2011043",
  "TYPECODE": "1",
  "BLU_ARRIVALCODE": "9",
  "REFERENCE": null,
  "DETAILS": null,
  "ORDERDO_SUBFORM": [
    {
      "TEXT": null,
      "TEXTLINE": 0
    }
  ],
  "BLU_ORDERSCONN_SUBFORM": [
    {
      "WEBQPRICE": 14885.78,
      "BLU_NAME": "Moran"
    }
  ],
  "ORDERITEMS_SUBFORM": [
    {
      "PARTNAME": "OCD0180",
      "TQUANT": 400,
      "CPRICE": 1.14,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "OCD0500",
      "TQUANT": 800,
      "CPRICE": 0.66,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "OCD0120",
      "TQUANT": 400,
      "CPRICE": 1.224,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE658A",
      "TQUANT": 60,
      "CPRICE": 48.90822,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE178A",
      "TQUANT": 174,
      "CPRICE": 1.67412,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE171A",
      "TQUANT": 174,
      "CPRICE": 1.67412,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE645A",
      "TQUANT": 36,
      "CPRICE": 48.90822,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE648A",
      "TQUANT": 42,
      "CPRICE": 48.90822,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BCE635A",
      "TQUANT": 36,
      "CPRICE": 48.90822,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BEF000R",
      "TQUANT": 500,
      "CPRICE": 2.52,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BKCS52W",
      "TQUANT": 20,
      "CPRICE": 7.62,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BKCS53W",
      "TQUANT": 20,
      "CPRICE": 7.62,
      "BLU_ACCOMPFLAG": "Y"
    },
    {
      "PARTNAME": "BKCS50G",
      "TQUANT": 20,
      "CPRICE": 7.62,
      "BLU_ACCOMPFLAG": "Y"
    }
  ]
}
```

Url	Client IP
/odata/Priority/tabula.ini/zbluran/ORDERS	212.150.105.21

Method	Server IP
POST	192.168.253.21

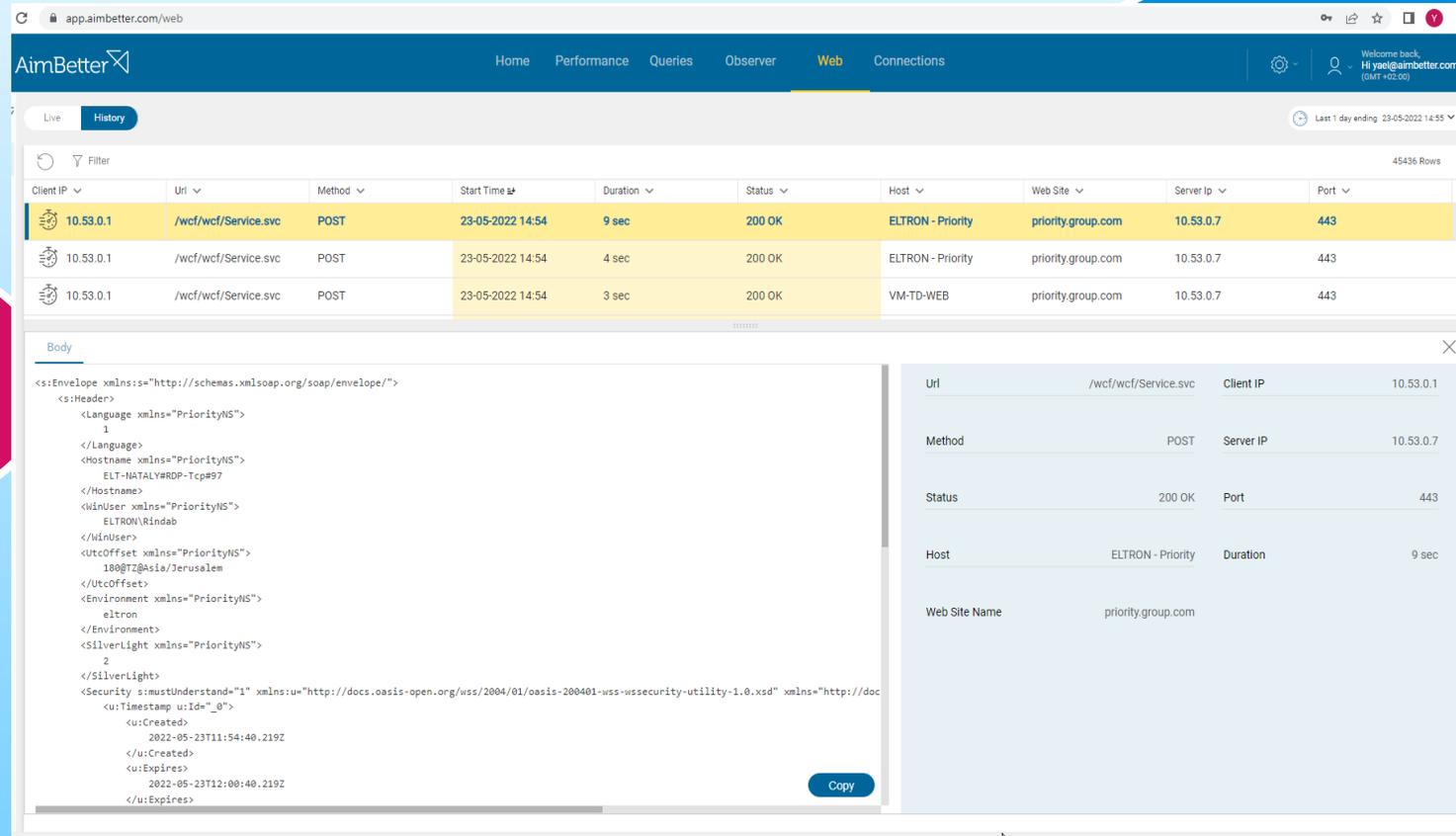
Status	Port
201 Created	443

Host	Duration
IIS2012	2 min

Web Site Name	Web Site
	207.232.54.210

When clicking on the record, AimBetter displays all the parameters involved in the request (marked):

AimBetter allows you to see requests that took place simultaneously and know for how long they were sharing the same timeframe:



The screenshot displays the AimBetter web application interface. At the top, there's a navigation bar with 'Home', 'Performance', 'Queries', 'Observer', 'Web', and 'Connections'. The 'Web' tab is active. Below the navigation bar, there's a 'Live' and 'History' toggle, and a 'Filter' button. The main area shows a table of requests with columns: Client IP, Url, Method, Start Time, Duration, Status, Host, Web Site, Server IP, and Port. The first row is highlighted in yellow. Below the table, there's a 'Body' section showing the XML content of the selected request. To the right of the XML, there's a metadata table with key-value pairs.

Client IP	Url	Method	Start Time	Duration	Status	Host	Web Site	Server IP	Port
10.53.0.1	/wcf/wcf/Service.svc	POST	23-05-2022 14:54	9 sec	200 OK	ELTRON - Priority	priority.group.com	10.53.0.7	443
10.53.0.1	/wcf/wcf/Service.svc	POST	23-05-2022 14:54	4 sec	200 OK	ELTRON - Priority	priority.group.com	10.53.0.7	443
10.53.0.1	/wcf/wcf/Service.svc	POST	23-05-2022 14:54	3 sec	200 OK	VM-TD-WEB	priority.group.com	10.53.0.7	443

```
<?xml version='1.0' encoding='utf-8'>
<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
  <s:Header>
    <Language xmlns='PriorityNS'>
      1
    </Language>
    <Hostname xmlns='PriorityNS'>
      ELT-NATALYA-RDP-Tcp#97
    </Hostname>
    <WUser xmlns='PriorityNS'>
      ELTRON/Rindab
    </WUser>
    <UtcOffset xmlns='PriorityNS'>
      18@TZ@Asia/Jerusalem
    </UtcOffset>
    <Environment xmlns='PriorityNS'>
      eltron
    </Environment>
    <SilverLight xmlns='PriorityNS'>
      2
    </SilverLight>
    <Security:s:mustUnderstand='1' xmlns:u='http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd' xmlns='http://doc
    <u:Timestamp u:Id='_0'>
      <u:Created>
        2022-05-23T11:54:40.219Z
      </u:Created>
      <u:Expires>
        2022-05-23T12:00:40.219Z
      </u:Expires>
    </u:Timestamp>
  </s:Header>
  <s:Body>
    <!-- Request Body Content -->
  </s:Body>
</s:Envelope>
```

Url	/wcf/wcf/Service.svc	Client IP	10.53.0.1
Method	POST	Server IP	10.53.0.7
Status	200 OK	Port	443
Host	ELTRON - Priority	Duration	9 sec
Web Site Name	priority.group.com		

The body displayed is of the highlighted request.

Being able to view the code behind each request is very useful for web developers, as they can modify their code to prevent the most common errors encountered by their users which they get aware of based on AimBetter's feedback. They don't need to wait until a user reports an error, as all errors are automatically logged by AimBetter and handled as soon as they're noticed.